Andreas Vogelsang, Kerstin Hartig, Florian Pudlitz, Aaron Schlutter, Jonas Winkler

Supporting the Development of Cyber-Physical Systems with Natural Language Processing: A Report

Vogelsang, Andreas; Hartig, Kerstin; Pudlitz, Florian; Schlutter, Aaron; Winkler, Jonas (2019): Supporting the Development of Cyber-Physical Systems with Natural Language Processing: A Report. Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, Live Studies Track, and Poster Track. (CEUR workshop proceedings ; 2376). Aachen: RWTH.

Supporting the Development of Cyber-Physical Systems with Natural Language Processing: A Report

Andreas Vogelsang, Kerstin Hartig, Florian Pudlitz, Aaron Schlutter, Jonas Winkler
Automated Systems Engineering Technologies (ASET)
Technische Universität Berlin, Germany
{*firstname.lastname*}@tu-berlin.de

Abstract

Software has become the driving force for innovations in any technical system that observes the environment with different sensors and influence it by controlling a number of actuators; nowadays called Cyber-Physical System (CPS). The development of such systems is inherently inter-disciplinary and often contains a number of independent subsystems. Due to this diversity, the majority of development information is expressed in natural language artifacts of all kinds. In this paper, we report on recent results that our group has developed to support engineers of CPSs in working with the large amount of information expressed in natural language. We cover the topics of automatic knowledge extraction, expert systems, and automatic requirements classification. Furthermore, we envision that natural language processing will be a key component to connect requirements with simulation models and to explain tool-based decisions. We see both areas as promising for supporting engineers of CPSs in the future.

1 Team Overview and Application Domain

The Automated Systems Engineering Technologies (ASET) group at the Technical University of Berlin is researching and developing technologies to support system engineers and automate time-consuming or error-prone tasks and process steps. With our research, we aim at the development of software-intensive systems that constantly observe their environment with different sensors and try to influence the environment in a desired way by controlling a number of actuators. Since software is becoming the most important and most critical part of these systems, they are now often called Cyber-Physical Systems (CPS) [Lee08].

Although software is becoming most critical for CPSs, their development is inherently inter-disciplinary in terms of the involved application domains (e.g., smart mobility) and the involved engineering disciplines (e.g., mechanics, electronics, and software). Due to this diversity, the majority of development information is expressed in natural language because NL can be read and understood by engineers and stakeholders independent of their background knowledge. In addition, the development of CPSs is driven by strong safety and security constraints because most of the times, humans or physical assets are impacted by the behavior of a CPS. CPS relevant development information expressed in natural language does not only include requirements but also safety analyses and assessments, architectural descriptions, test cases, and many more. Development information is often spread over hundreds of documents with thousands of single entries. For example, the specification repository of a telematics system of a modern automotive system that we are analyzing contains 28,867 documents with 2,423,624 entries. On the other hand, most of the engineering tasks for CPS are performed manually by experts who make heavy use of their experience and domain expertise. These experts must be supported to cope with the amount and richness of information expressed in natural language.

We try to tackle these challenges in our group by developing three areas of competence: Artificial Intelligence for Systems Engineering, Model-based Engineering, and Validation by Simulation. We do research in an application-oriented manner and test our technologies continuously in practice.[1]

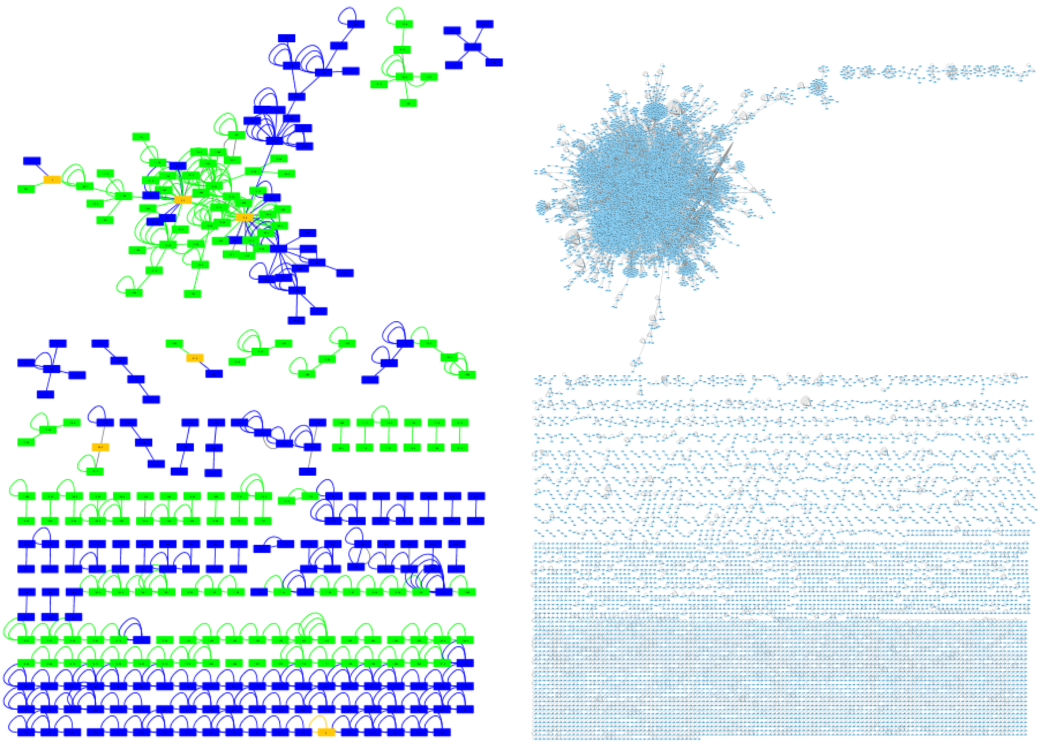
2 Past and Current Research on NLP for CPS Development

We use NLP techniques to automatically extract specific information from large corpora of textual documents, develop expert systems that can be used to retrieve answers to specific queries, and to classify information in textual documents automatically.

2.1 Automatic Knowledge Extraction

Engineers of CPSs are challenged by comprehending the concepts mentioned in a requirement because coherent information is spread over several requirements documents. The reasons are that single documents often only cover the view of one discipline (e.g., mechanics or software) or that the mentioned concepts strongly depend on other parts of the system that are described in another document (cf. [VF13]).

We have developed a natural language processing pipeline to transform a set of heterogeneous natural language requirements from different documents into a knowledge representation graph [SV18]. The graph provides an orthogonal view onto the concepts and relations written in the requirements. In a first validation of the approach, we applied it to two separate requirements documents including more than 7,000 requirements from industrial systems (see Figure 1). As the first requirements document included several subsystems, we were able to analyze which concept descriptions are distributed over subsystems and where those subsystems had intersections to each other (see Figure 1a).



(a) Exterior lighting and adaptive cruise control (b) Charging system for electric vehicles

Figure 1: Knowledge representation graphs extracted from two requirements documents

[1]`https://aset.tu-berlin.de`

A second area that we have worked on is the extraction of terms that should be defined and clarified in an inter-disciplinary project (i.e., creating a glossary). Creating glossaries for large corpora of textual documents is important for creating a shared understanding between all engineers and for uncovering potential sources of ambiguity (cf. [FEG18]). However, creating glossaries is also an expensive task because it is largely manual. Automatic glossary term extraction methods often focus on achieving a high recall rate and, therefore, favor linguistic processing for extracting glossary term candidates and neglect the benefits from reducing the number of candidates by statistical filter methods [ASBZ17]. However, especially for large datasets, a reduction of the likewise large number of candidates may be crucial.

We have demonstrated how to automatically extract relevant domain-specific glossary term candidates from a large body of requirements, the CrowdRE dataset [GCKV18]. Our hybrid approach combines linguistic processing and statistical filtering for extracting and reducing glossary term candidates. In a twofold evaluation, we examined the impact of our approach on the quality and quantity of extracted terms. We showed that a substantial degree of recall can be achieved even if we applied statistical filters to reduce the number of false positives. Furthermore, we advocate requirements coverage as an additional quality metric to assess the term reduction that results from our statistical filters. Results indicate that with a careful combination of linguistic and statistical extraction methods, a fair balance between later manual efforts and a high recall rate can be achieved.

2.2 Expert Systems

The development of CPSs must often adhere to development standards to ensure certain non-functional properties (e.g., ISO 26262 for safety-critical systems in automotive). According to the standard, the hazard analysis and risk assessment (HARA) is one of the first safety activities during the development of safety-related systems. In this analysis, experts examine potential malfunctions and their consequences in different situations, and specify safety goals to reduce risks to an acceptable level. Performing HARAs is a time-consuming and expensive activity because it is expert-driven and requires extensive experience and domain knowledge. Thus, domain experts would benefit from decision support that allows the automated reuse of approved knowledge from previous analyses. However, automated knowledge reuse is considered a challenging task.

We have developed an information retrieval system that represents the results from previous HARAs in a semantic network and searches it for useful recommendations during a new HARA by applying spreading activation algorithms [HK16]. We use the underlying data model of the HARA document to automatically create a basic semantic networks from semi-structured HARA documents. Natural language processing techniques help us to refine the networks and extract semantics from coarse-grained text fragments such as *description* elements. Our approach aims at making optimal use of the reuse potential and, therefore, increasing the consistency of HARAs and the efficiency of their development. In an evaluation, we have implemented the approach based on a set of 155 existing HARA documents. The evaluation reveals good quality of the retrieval results and indicates, which configuration settings are advantageous. Moreover, we showed how configuration settings can be optimized with evolutionary algorithms, which extends the developer's tool set.

2.3 Automatic Requirements Classification

In CPS development, requirements are not only used to describe the intended characteristics of the envisioned system but also for a number of management tasks such as effort estimation, test planning, or contract design. For these tasks, it is important to assess and classify single requirements (e.g., by priority, estimated effort, potential verification method, etc.) In single specifications from the automotive domain, we have seen up to 6,048 attributes with partly more than 100 different attribute entries, which where used to annotate requirements in documents.

We have developed an automatic classification approach for textual requirements that can be used to support quality assurance. The approach uses word embeddings to encode texts and convolutional neural networks to assign membership values to predefined classes [WV16]. After talking to engineers, we have instantiated the approach for important attributes. One example is the classification of textual entries into the classes *requirement* and *information*. While requirements are legally binding, information entries contain additional content such as explanations, summaries, or figures. Our approach is able to detect errors in this attribute with a recall of 0.95 and a precision of 0.30.

3 Future Research on NLP for CPS Development

We envision that natural language processing will be a key component to connect requirements with simulation models and to explain tool-based decisions. We see both areas as promising for supporting engineers of CPSs in the future.

3.1 Connecting NL Requirements and Simulation

CPSs are complex because they are often assembled from a number of systems that interact independently to some degree. In such a context, formal reasoning about resulting system behavior is hard or even impossible. Simulation is often a better alternative to explore the complex interplay of systems. However, currently, simulation in practice is either used in the very early stages for feasibility studies or in the very late stages to test the implemented system. Requirements engineers do not profit from simulation results because the simulations are not connected to the requirements in the specifications.

We aim at closing this gap by giving requirements engineers the possibility to relate natural language requirements with observable events in simulators. As a result, the requirements engineer receives information that annotate the requirements with results from multiple simulation runs. We present a first prototype of this approach in this year's REFSQ conference [PV19]. The challenge is to make the mapping process as easy and convenient as possible for the requirements engineer such that the effort pays off for him or her. We aim at using NLP to support this process (e.g., by giving recommendations based on similarity measures between requirements and descriptions of simulation events).

3.2 Explainability

In many cases, the purpose of addressing RE tasks with NLP techniques is to support the human analyst and not completely replace him or her. Therefore, it is becoming more and more important that tool results go along with explanations of the results. Sometimes, the explanation is even more helpful than the actual result. However, especially with the use of data-driven technologies such as machine learning, it is challenging to explain tool decisions.

We try to emphasize the importance of explainability and search for solutions in this field. One example is the automatic requirements classification tool that we already introduced in the previous section. To make the decisions of the tool explainable, we have developed a mechanism that traces back the decision through the neural net and highlights fragments in the initial text that influenced the tool to make its decision [WV17]. As shown in Figure 2, it appears that the word "must" is a strong indicator for a requirement, whereas the word "required" is a strong indicator for an information element. While the first is not very surprising, the latter could indicate that information elements often carry rationales (why something is *required*).

Classes	requirement information
requirement	the duration until the switch is recognized as hanging must be a configurable parameter .
information	the component conditionally drives an external fan . this fan is required for active ventilation of the headlight .

Figure 2: Automatic Classification of textual specification objects into classes *requirement* and *information*.

Another example in which we looked for explainability is in the recommendations from expert system. In Section 2.2, we introduced our expert system for hazard and risk analysis. In this approach, we used *spreading activation* as a technique to extract relevant concepts for a certain query. Spreading activation is a well-known semantic search technique to determine the relevance of nodes in a semantic network. When used for decision support, meaningful explanations of semantic search results are crucial for the user's acceptance and trust. Therefore, we have developed an approach that exploits the so-called spread graph, a specific data structure that comprises the spreading progress data [MH16]. We have shown how to retrieve the most relevant parts of a network by minimization and extraction techniques and formulate meaningful explanations.

4 Conclusions

In this report, we present past work and future research directions in the area of natural language processing in the Automated Systems Engineering Technologies (ASET) group at the Technical University of Berlin. With our research, we mainly target the development of cyber-physical systems (CPS). We argue that the majority of development information for CPSs is expressed in natural language due to the diversity in involved application domains and engineering disciplines. We have worked on using NLP techniques to extract specific information from large corpora of textual documents automatically, develop expert systems that can be used to retrieve answers to specific queries, and to classify information in textual documents automatically. We envision that natural language processing will be a key component to connect requirements with simulation models and to explain tool-based decisions. We see both areas as promising for supporting engineers of CPSs in the future.

References

[ASBZ17] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer. Automated extraction and clustering of requirements glossary terms. *IEEE Transactions on Software Engineering (TSE)*, 43(10), 2017.

[FEG18] A. Ferrari, A. Esuli, and S. Gnesi. Identification of cross-domain ambiguity with language models. In *International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, 2018.

[GCKV18] T. Gemkow, M. Conzelmann, K.Hartig, and A. Vogelsang. Automatic glossary term extraction from large-scale requirements specifications. In *26th IEEE International Requirements Engineering Conference (RE)*, 2018.

[HK16] K. Hartig and T. Karbe. Recommendation-based decision support for hazard analysis and risk assessment. In *8th International Conference on Information, Process, and Knowledge Management (eKNOW)*, 2016.

[Lee08] E. A. Lee. Cyber physical systems: Design challenges. In *11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008.

[MH16] V. Michalke and K. Hartig. Explanation retrieval in semantic networks – understanding spreading activation based recommendations. In *8th International Conference on Knowledge Discovery and Information Retrieval (KDIR)*, 2016.

[PV19] F. Pudlitz and A. Vogelsang. A lightweight multilevel markup language for connecting software requirements and simulations. In *25th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ)*, 2019.

[SV18] A. Schlutter and A. Vogelsang. Knowledge representation of requirements documents using natural language processing. In *1st Workshop on Natural Language Processing for Requirements Engineering (NLP4RE)*, 2018.

[VF13] A. Vogelsang and S. Fuhrmann. Why feature dependencies challenge the requirements engineering of automotive systems: An empirical study. In *21st IEEE International Requirements Engineering Conference (RE)*, 2013.

[WV16] J. Winkler and A. Vogelsang. Automatic classification of requirements based on convolutional neural networks. In *3rd International Workshop on Artificial Intelligence for Requirements Engineering (AiRE)*, 2016.

[WV17] J. Winkler and A. Vogelsang. What does my classifier learn? A visual approach to understanding natural language text classifiers. In *22nd International Conference on Natural Language & Information Systems (NLDB)*, 2017.